PASSED

Groupreport

Akshayaswuroupikka Balasubramanian,yixuan xu

February 19, 2016

Assignment 1: Cluster sampling

An opinion pool is assumed to be performed in several locations of Sweden by sending interviewers to this location. Of course, it is unreasonable from the financial point of view to visit each city. Instead, a decision was done to use random sampling without replacement with the probabilities proportional to the number of inhabitants of the city to select 20 cities. Explore the file population.xls

1.1

The necessary information is imported to R and the data looks like:

```
## Warning: package 'XLConnect' was built under R version 3.2.3
```

```
## Warning: package 'XLConnectJars' was built under R version 3.2.3
```

```
##         city Population
## 6  Botkyrka      81195
## 7  Danderyd      31150
## 8     Ekerö      25095
## 9   Haninge      76237
## 10 Huddinge      95798
## 11 Järfälla      65295
```

1.2

A function is written which selects 1 city from the whole list by the probability scheme offered above using uniform random number genarator is given below:

```
Random_city<-function(data){
  selected_city <-NULL
  ndata<-data
  for( i in 1:20){
    data$prob <- data$Population / sum(data$Population)
    cumpro<- cumsum(data$prob)
    r<- runif(1,0,1)
    selected_city[i]<-min(which(cumpro>=r))
    data=data[-min(which(cumpro >=r)),]
  }
  return(ndata[selected_city,])
}
```

1.3

The function you have created above is used to select one city and remove that city from the list.This function is applied again to the updated list of the cities until we get exactly 20 cities.

1.4

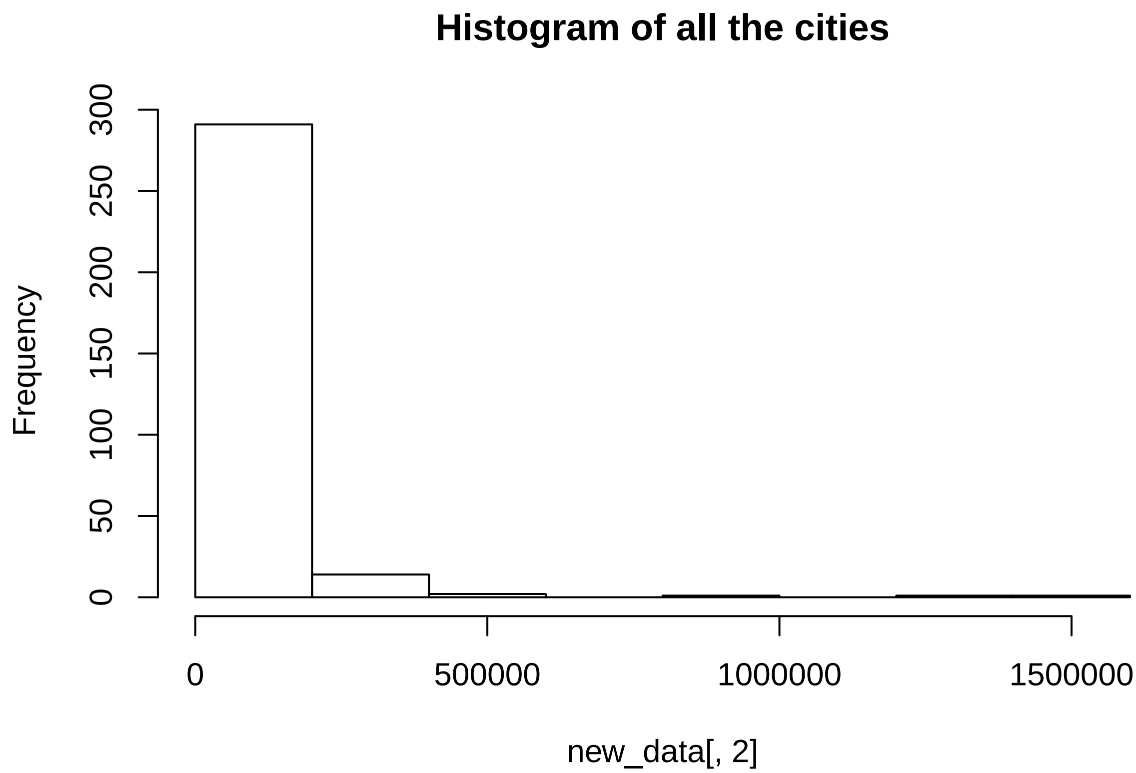
The following below are the selected cities:

```
##                city Population
## 38          Uppsala     194751
## 239      Surahammar       9980
## 142    Östra Göinge      13526
## 148          Laholm      23345
## 28       Vallentuna      29361
## 177         Mölndal      60381
## 119            Höör      15261
## 180           Skara      18455
## 20            Solna      66909
## 20.1          Solna      66909
## 105      Karlskrona      63342
## 110            Bjuv      14813
## 222       Karlskoga      29742
## 128        Perstorp       6983
## 93           Kalmar      62388
## 151             Ale      27394
## 10         Huddinge      95798
## 52          Boxholm       5248
## 223           Kumla      20214
## 240        Västerås     135936
```

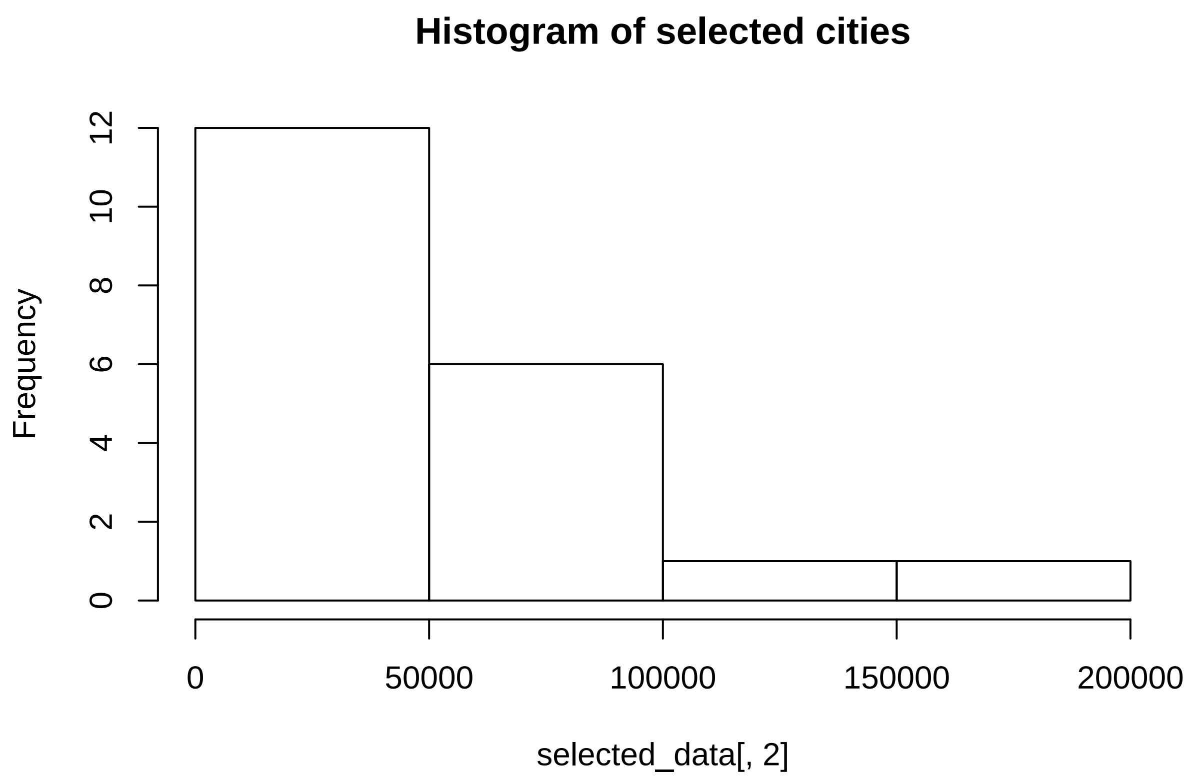
As we can see from the list most of the cities which are selected are high in population. (The result is different when we rerun the code, cause we did not set seed.)

1.5

The histogram showing the size of all cities of the country and another histogram showing the size of the 20 selected cities is plotted :



Histogram of all the cities
Frequency
0
50
100
150
200
250
300
0
500000
1000000
1500000
new_data[, 2]



Histogram of selected cities
Frequency
0
2
4
6
8
10
12
0
50000
100000
150000
200000
selected_data[, 2]

From the histogram of the two data set we could see that the distributions are almost similar.Since the probability of the city to be choosen depends on the population, the city with more population has higher pobability to be selected by uniform random. In this case, the sampling function could obtain the main information of the original data.

Assignment 2: Different distributions

2.1

The double exponential (Laplace) distribution is given by formula:

$$DE(\mu, \alpha) = \frac{\alpha}{2}\ exp(-\alpha\ |x - \mu|)$$

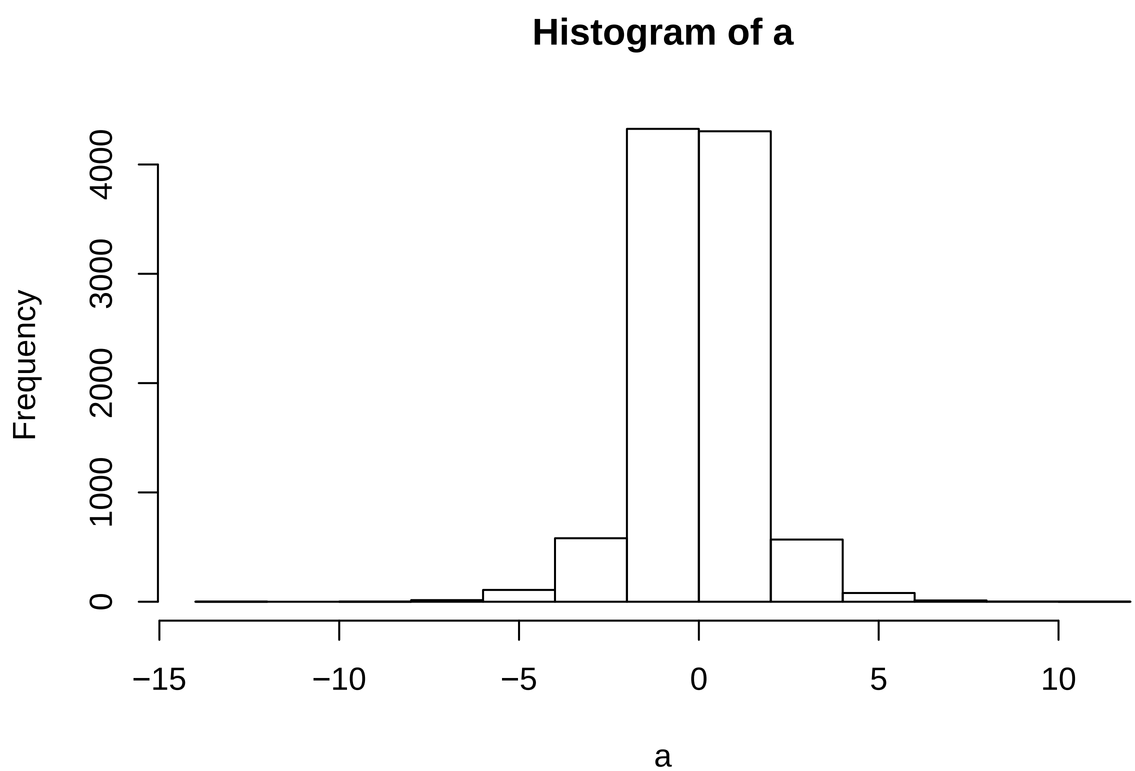
$$f_X(x) = \begin{cases} \frac{\alpha}{2} e^{(-\alpha(x-\mu))} & \text{if } x \geq \mu \\ \frac{\alpha}{2} e^{(-\alpha(\mu-x))} & \text{if } x < \mu \end{cases}$$

$$F_X(x) = \int_{-\infty}^{x} f(t)d(t) = \begin{cases} 1 - \frac{1}{2} e^{-\alpha(x-\mu)} & \text{if } x \geq \mu \\ \frac{1}{2} e^{-\alpha(x-\mu)} & \text{if } x < \mu \end{cases}$$

$$F^{-1}(y) = x = \begin{cases} \mu - \frac{ln(2-2y)}{\alpha} & \text{if } y \geq 0.5 \\ \mu + \frac{ln(2y)}{\alpha} & \text{if } y < 0.5 \end{cases}$$

Set DB(0,1), then transform from U to X:

$$X = \begin{cases} ln(2u) & \text{if } u \geq 0.5 \\ -ln(2 - 2u) & \text{if } u < 0.5 \end{cases}$$



From the above plot, it is easy to see that the result looks reasonable. The Laplace distribution has fatter tails than the normal distribution. The probability density function of the Laplace distribution is also reminiscent of the normal distribution; however, whereas the normal distribution is expressed in terms of the squared difference from the mean μ, the Laplace density is expressed in terms of the absolute difference from the mean.

2.2

Step 1: Generate Y from U[0,1] use

$$F^{-1}(y) = x = \begin{cases} \mu - \frac{ln(2-2y)}{\alpha} & \text{if } y \geq 0.5 \\ \mu + \frac{ln(2y)}{\alpha} & \text{if } y < 0.5 \end{cases}$$

Step 2: Generate U from U[0,1].

Step 3: If $U \leq \frac{f_X(Y)}{cf_Y(Y)}$ take Y else return to step 1.

$$f_X(Y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$f_Y(Y) = \frac{\alpha}{2} e^{-\alpha|x-\mu|}$$

$$U \leq \frac{f_X(Y)}{cf_Y(Y)}$$

Set DE(0,1) for $f_Y(Y)$, N(0,1) for $f_X(Y)$.

$$f_X(Y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

$$f_Y(Y) = \frac{1}{2} e^{-|x|}$$

$$h(x) = \frac{f_X(Y)}{f_Y(Y)} = \sqrt{\frac{2}{\pi}} e^{(|x| - \frac{x^2}{2})}$$

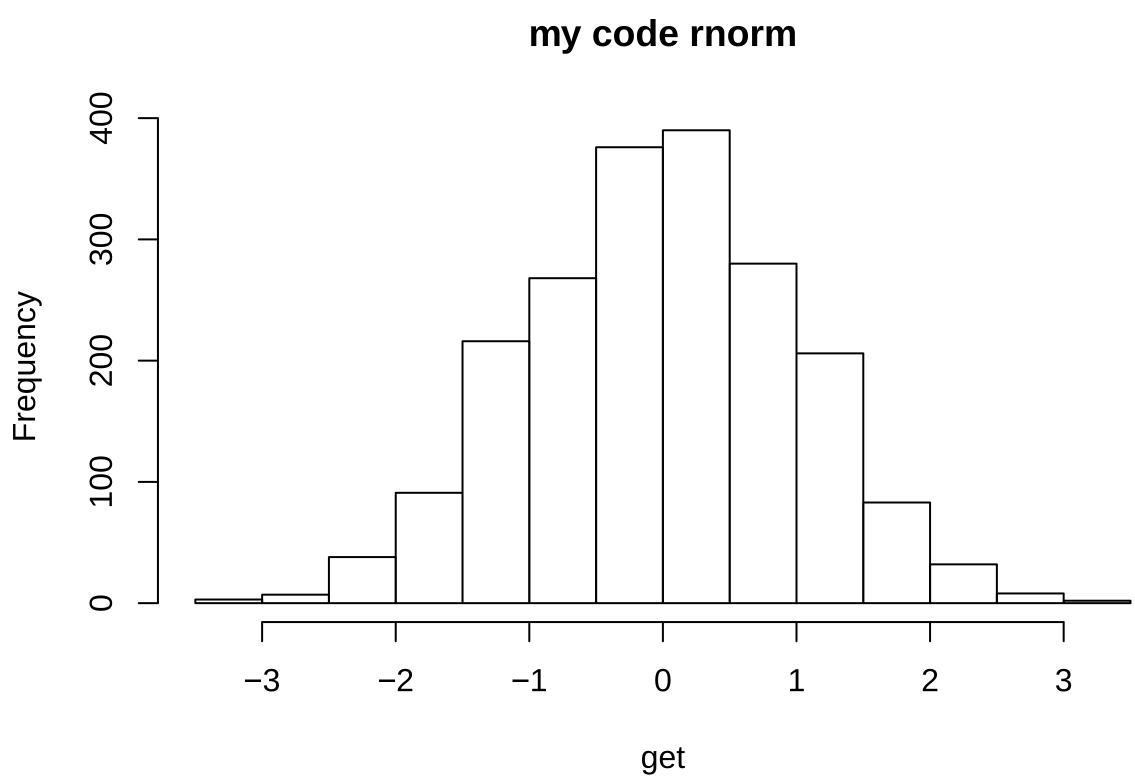
Compute its maximum ($h'(x) = 0$), so $e^{(|x| - \frac{x^2}{2})}$ have maximum value, thus $x = 1$ or -1

$$c f_Y(x) \geq f_X(x)$$

$$c = \sqrt{\frac{2e}{\pi}} = 1.3154892$$

And

$$\frac{f_X(Y)}{c f_Y(Y)} = e^{-\frac{(|Y|-1)^2}{2}}$$



average rejection rate:

```
## [1] 0.2383854
```

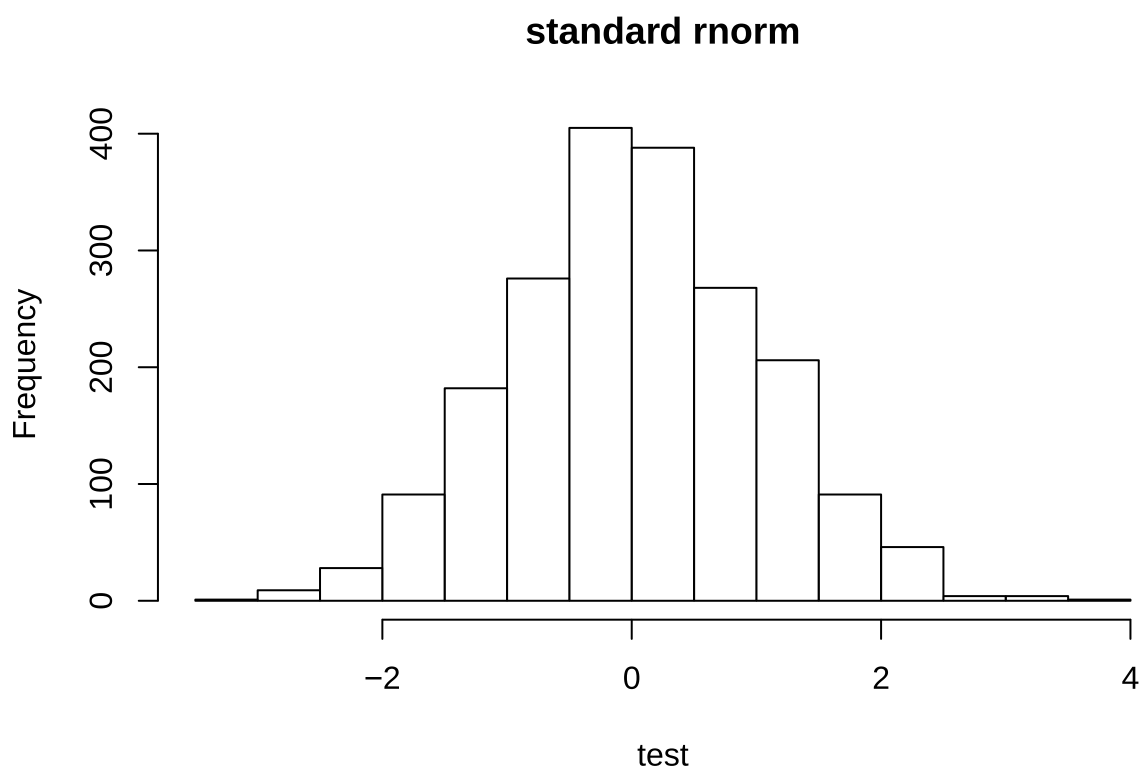
$$
\begin{aligned}
p &= \int_{\infty}^{-\infty} \frac{f_X(y)}{cf_Y(y)} \times f_Y(y)\mathrm{d}y \\
&= \frac{1}{c}\int_{\infty}^{-\infty} f_X(y)\mathrm{d}y \\
&= \frac{1}{c}
\end{aligned}
$$

expected rejection rate should be $1 - \frac{1}{c}$

expected rejection rate:

```
## [1] 0.2398265
```

The R and ER looks nearly the same.



The two histograms obtained are almost the same. That we can use our function instead of rnorm().

contribution

We discussed the results of our individual reports.The first part of the assignment contains akshaya's(code and text) and the second part contains yixuan's(code and text).

Appendix - R-code

```
## ---- echo=FALSE, message=FALSE------------------------------------------
library(XLConnect)
data=loadWorkbook("population.xls")
data_frame=readWorksheet(data, sheet = "Table", header = FALSE)
new_data=data_frame[6:315, 2:3]
names(new_data) <- c( "city", "Population")
new_data$Population=as.numeric(new_data$Population)
probability <- new_data$Population / sum(new_data$Population)
head(new_data)

## ------------------------------------------------------------------------
Random_city<-function(data){
  selected_city <-NULL
  ndata<-data
  for( i in 1:20){
    data$prob <- data$Population / sum(data$Population)
    cumpro<- cumsum(data$prob)
    r<- runif(1,0,1)
    selected_city[i]<-min(which(cumpro>=r))
    data=data[-min(which(cumpro >=r)),]
  }
  return(ndata[selected_city,])
}

## ---- echo=FALSE---------------------------------------------------------
selected_data=Random_city(new_data)

## ---- echo=FALSE---------------------------------------------------------
selected_data

## ---- echo=FALSE---------------------------------------------------------
hist(new_data[,2],main="Histogram of all the cities")
hist(selected_data[,2],main="Histogram of selected cities")

## ---- echo=FALSE---------------------------------------------------------
u <- runif(10000,0,1)
ee <- function(u, mu, alpha){
  n <- length(u)
  result <- NULL
  for(i in 1:n){
    if(u[i] >= 0.5){
      result[i] <- mu-(log(2-2*u[i]))/alpha
    }else{
      result[i] <- mu + (log(2*u[i]))/alpha
    }
  }
  return(result)
}
a <- ee(u,0,1)
hist(a)
```

```r
## ---- echo=FALSE--------------------------------------------------------
ar <- function(c){
  result <- c()
  i <- 0
  repeat{
    u1 <- runif(1,0,1)
    a1 <- ee(u1,0,1)
    u2 <- runif(1,0,1)
    i <- i+1
    if(u2 <= (exp(-a1^2/2)/sqrt(2*pi))/(c*exp(-abs(a1))/2)){
      result[i] <- a1
    }else{
      result[i] <- NA
    }
    if(length(result[-which(is.na(result))])==2000){
      break
    }
  }
  return(result)
}

## ---- echo=FALSE--------------------------------------------------------
c <- sqrt(exp(1)*2/pi)

## ---- echo=FALSE--------------------------------------------------------
get <- ar(c)
get1 <- get[-which(is.na(get))]
hist(get, main = "my code rnorm")

## ---- echo=FALSE--------------------------------------------------------
1-length(get1)/length(get)

## ---- echo=FALSE--------------------------------------------------------
1-1/c

## ---- echo=FALSE--------------------------------------------------------
test <- rnorm(2000)
hist(test, main = "standard rnorm")

## ----code=readLines(knitr::purl("GroupReport.Rmd", documentation = 1)), eval = FALSE----
## NA
```